

EXHIBIT C

05/06/02



1049 U.S. PTO

CAHILL, SUTTON & THOMAS P.L.C.

155 Park One
2141 East Highland Avenue
Phoenix, Arizona 85016
Ph. (602) 956-7000
Fax (602) 495-9475

1036 U.S. PTO

10/139924



05/06/02

Docket No. 6339-A-1Date May 6, 2002

Express Mail Mailing Label No. 816743966 US

Date of Deposit: May 6, 2002

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to Box PATENT APPLICATION, Commissioner for Patents, Washington, D.C. 20231.

Marvin A. Glazer
Date

Box PATENT APPLICATION
Commissioner for Patents
Washington, D.C. 20231

Sir:

Transmitted herewith for filing is the patent application of

Inventors: **LEON E. HAUCK and BRENT J. BURVAL**For: **METHOD FOR RESTRICTING ACCESS TO A WEB SITE
BY REMOTE USERS**

Enclosed are:

- X Four sheets of informal drawings
- X Declaration and Power of Attorney
- X Specification (including claims and abstract) 25 pages

X Assignment

X Form PTO-1619A Recordation Form Cover Sheet

CLAIMS AS FILED

Total
Claims 16 - 20 = -0- x \$ 9.00 = \$ -0-

Independent
Claims 2 - 3 = -0- x \$42.00 = \$ -0-

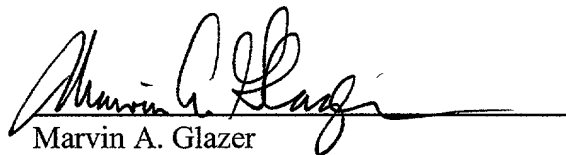
Basic Fee: \$370.00

Total Filing Fee: \$370.00

X The Commissioner is hereby authorized to charge any additional fees which may be required by this paper to Deposit Account No. 03-0088 and to credit any overpayment to this Deposit Account. Two (2) duplicate copies of this sheet are enclosed.

X A check in the amount of **\$410.00** to cover the filing fee and the recordation fee is enclosed.

X Applicant claims small entity status. See 37 C.F.R. 1.27.


Marvin A. Glazer
Reg. No. 28,801

**METHOD FOR RESTRICTING ACCESS TO
A WEB SITE BY REMOTE USERS**

Background of the Invention

1. Field of the Invention:

The present invention relates generally to a client/server authentication system used with Internet or intranet Web sites and/or Web pages, mail servers, or file transfer protocol (FTP) servers, and more particularly, to such a client/server authentication system that ensures that the client is operating from a specific, pre-authorized client machine.

2. Description of the Related Art:

There has long been recognized a need for increasing the security of Internet and intranet Web sites and/or selected Web pages accessible via such Web sites. In some instances, access must be restricted to particular authorized employees due to the confidential and sensitive nature of information stored on the server computer; this is likely the case in many corporate intranet systems, in which travelers and tele-commuters need to log onto the intranet from remote locations. In other cases such as subscription-based Web sites, the data stored on the server has commercial value, and the owner of the Web site limits access to paying subscribers. In still other instances, such as on-line banking, access to portions of a financial institution's Web site must be limited to the authorized owner of a particular bank account.

Various security measures have been employed in the past to restrict the access of a client to data stored on a remote server. For example, it is common for Web site operators to save authorized user names and passwords, and to require the user to "log-in" by entering his or her user name and password in order to access information stored on the server. In other cases, the user must enter a key code assigned to the user. However, these security methods are easily defeated, as the user can provide his or her user name, password and/or key code to third parties, thereby allowing others to gain access to the "secured site" from virtually any personal computer. Likewise, if the user name and password and/or key code are stolen, then the thief can gain access to the system, at least until the user reports the theft of such information.

1 Another type of security measure that has been employed is a hardware-lock component,
2 sometimes called a “dongle”, which must be plugged into a port of the user’s computer; the
3 dongle provides a code that can be read by the user’s computer and transmitted to the server
4 computer to verify the identity of the user. Once again, however, such security measures are
5 defeated if the owner of the hardware-lock component loans it to third parties, who can then
6 access the restricted data from any available computer. In addition, such “dongles” are often
7 lost or stolen, posing a substantial inconvenience to genuine authorized users, and creating
8 additional opportunities for security breaches.

9 Accordingly, it is an object of the present invention to provide a client/server
10 authentication system which provides an increased level of security.

11 Another object of the present invention is to provide such an authentication system
12 which does not require the user to possess any special hardware components, or “dongles”, in
13 order to gain access to the server.

14 Still another object of the present invention is to provide such an authentication system
15 capable of restricting a client to a particular personal computer or other client machine in order
16 to gain access to a Web site and/or Web pages on an Internet and/or intranet site.

17 Yet another object of the present invention is to provide such an authentication system
18 that easily allows for multiple levels of access wherein some authorized users have greater
19 access to Web pages than other authorized users.

20 An additional object of the present invention is to provide such an authentication system
21 that eliminates the need for the server administrator to store, or administer, user names,
22 passwords or other user authentication data.

23 A further object of the present invention is to provide such an authentication system
24 which avoids the necessity for a user to remember a user name, password and/or key code in
25 order to gain authorized access to a protected site.

26 These and other objects of the present invention will become more apparent to those
27 skilled in the art as the description of the present invention proceeds.

28

1 Summary of the Invention

2 Briefly described, and in accordance with a preferred embodiment thereof, the present
3 invention relates to a method of restricting access to data maintained on a server computer by an
4 authorized client machine connected by a network to the server computer; often, the client
5 machine is located remotely from the server computer. The novel method includes the step of
6 installing a client-side software program on the client machine for generating a client machine-
7 specific identifier. In the preferred embodiment of the present invention, this client-side
8 software program is either downloaded from a remote computer (which could be the server
9 computer that hosts the protected Web site or another remote computer) or installed from digital
10 media (CDROM, DVD, floppy disk, etc.). The client machine-specific identifier is determined
11 by identifying particular characteristics of the particular computer or other client machine upon
12 which such client-side software program is installed. The client-side software program is
13 operated on the client machine to generate the aforementioned machine-specific identifier. This
14 machine-specific identifier is passed to a remote computer, or perhaps to a person, for
15 generating a unique password based upon the machine-specific identifier; the unique password
16 is then provided to the user of the client machine for use by the client-side software program.
17 The remote computer used to generate the unique password may be a server computer that hosts
18 a protected Web site, or it may be another computer remote from the client machine.

19 The method of the present invention preferably includes the step of installing a server-
20 side software program on a server computer for generating the unique password described
21 above. This server computer could be the same server computer that hosts the protected Web
22 site, but could also be a different server computer for enhanced security. The client machine-
23 specific identifier is transmitted to such server computer, and the server-side software program,
24 installed on the server computer, generates the unique password. The server-side software
25 program performs an algorithm for generating the unique password based upon the client
26 machine-specific identifier generated by the client machine.

27 When a user desires access to data maintained on the server computer, the client-side
28 software re-generates its machine-specific identifier. A comparison is then made, preferably by

1 the client-side software itself, to compare the re-generated machine-specific identifier to the
2 unique password saved by the user in the client machine. If the client-side software verifies that
3 the re-generated machine-specific identifier properly corresponds with the unique password
4 previously provided to such client-side software, then the client-side software recognizes the
5 client machine as being authorized to access data maintained on the server computer. On the
6 other hand, if the client-side software is unable to verify that the re-generated machine-specific
7 identifier properly corresponds with the unique password previously provided to such client-side
8 software, then the client-side software refuses to recognize the client machine as being
9 authorized to access data maintained on the server. While it is preferred that the above-
10 described comparison/verification process is performed at the client machine, the
11 comparison/verification process can also be performed by the server computer that the user is
12 trying to access.

13 Another aspect of the present invention relates to the use of a session identifier to control
14 access to the protected data during a particular access session by an authorized client machine.
15 In the example of a protected Web site served by Microsoft Internet Information Server (IIS®)
16 software, a session identifier is created by the server computer each time a networked computer
17 first requests access to such site for the purpose of indicating a particular working session for
18 the requesting machine. In the case of known FTP servers, session IDs might not be created
19 automatically; in this case, the server-side software program mentioned above includes a plug-in
20 to the existing FTP server software for causing such a session identifier to be created. The
21 session identifier of an authorized client machine is stored in the server computer, or in another
22 computer remote from the client machine, in a temporary storage table along with additional
23 data, such as an access level code, machine specific identifier, etc. In addition, the session
24 identifier information is transmitted from the server computer to the client machine in the form
25 of "cookie" data (in the example of protected HTTP protocol Web sites), and the client machine
26 stores this transmitted "cookie" data containing the session identifier information.

27 Thereafter, each time the client machine requests access to the protected data during the
28 same access session, the client machine sends the saved session identifier back to the server

1 computer, as by including such session identifier information within the heading of the request.
2 Each such request is transmitted in the form of an HTTP "header" that typically includes the site
3 address (URL), previously stored cookie information particular to the user, and the session
4 identifier. The server computer then compares the session identifier sent by the client machine
5 with session identifiers already stored in the server computer's temporary storage table to
6 determine whether a correspondence exists therebetween. If a correspondence exists between
7 the session identifier sent by the client machine with session identifiers stored in the server
8 computer temporary storage table, then the requested access is allowed. On the other hand, if
9 there is no correspondence between the session identifier sent by the client machine with session
10 identifiers already stored in the server computer's temporary storage table, then the requested
11 access is denied. In this manner, a particular authorized client machine is allowed continued
12 access to the protected data during the particular working session indicated by the session
13 identifier. This method provides the advantage that the server computer need not maintain a
14 permanent storage table of all authorized client machines.

15 The ability of the server computer to use only a temporary table of authorized users who
16 are currently logged-on greatly simplifies the use of multiple servers that collectively serve up a
17 protected site. For example, some Web sites are so large, have so many subscribers, or host so
18 much content, that they require tens or hundreds of server computers to handle, and balance, the
19 work load. If each of these tens, or hundreds, of server computers had to maintain a permanent
20 table of authorized users, then significant administrative maintenance effort would be required
21 to keep each such table up to date. In contrast, with the present invention, each such server
22 would, at most, temporarily store the session identifiers of only those users currently engaged in
23 an active session with such server. Administrative overhead can further be reduced by
24 maintaining the temporary session ID table on a single master server, and allowing the other
25 tens or hundreds of servers to query the temporary session ID table maintained on the master
26 server to verify that a particular user request is authorized.

27 Another aspect of the present invention relates to providing different levels of access to
28 different authorized client machines. For example, some users may be authorized to access all

1 pages of a protected Web site, while other users may be limited to only lower (i.e., less secure,
2 more public) levels of access to the protected data. While such site specific information could
3 be maintained on a permanent basis on the server computer, this task would add to the
4 complexity of the server-side software and maintenance. In the preferred embodiment of the
5 present invention, the client-side software program installed on the client machine determines
6 which areas, or sites, of the protected data residing on the server computer can be accessed by
7 the particular client machine, as determined by its comparison of the password previously
8 entered by the user versus the machine-specific identifier generated by the client-side software
9 program. Such comparison can provide additional information beyond merely whether access is
10 to be granted or denied; for example, this comparison can return information indicating the level
11 of service to which a particular client is entitled, or whether the client's subscription to such
12 service has already expired.

13 As mentioned, the client-side software program can generate access level information to
14 indicate the particular data area(s) of the server computer which the client machine is authorized
15 to access. Assuming that access was granted, then this access level information is transmitted
16 by the client machine to the server computer's temporary storage table along with the initial
17 request for access to the protected data during each access session. In response to a request to
18 access protected data, the server computer compares the session identifier contained in the
19 header of the request to the session identifier information that was saved in the server
20 computer's temporary storage table during the initial request for access. The server computer
21 determines whether a correspondence exists between the session identifier that was included in
22 the header of the request for access sent by the client machine versus the session identifiers
23 already stored temporarily in the server computer storage table. If so, then the temporary
24 storage table can provide access level information which can be used by the server to control
25 such client machine's access to only those levels of service to which such client machine is
26 authorized. The server computer permits a client machine to access authorized levels of the
27 requested site during a particular user access session if 1) the session identifier sent by the client
28 machine matches the session identifier information already stored in the server computer's

1 temporary storage table for such client machine, and 2) the access level requested by the client
2 machine matches, or is at a "lower" level than, the access level already stored in the temporary
3 storage table for such client machine; otherwise, the requested access is denied.

4
5 Brief Description of the Drawings

6 Fig. 1 is a block diagram showing a computer network connecting a series of client
7 machines to a server computer.

8 Figs. 2A, 2B, and 2C collectively form a flowchart that illustrates the steps performed by
9 a preferred embodiment of the present invention.

10
11 Detailed Description of the Preferred Embodiment

12 The present invention relates to a client/server user authentication system used with
13 Internet or intranet Web sites and/or Web pages that require a higher degree of security than
14 typical static or dynamic username/password methods offer. The present invention is intended
15 for server administrators who require absolute control over the access rights of client-machines
16 connecting to their server(s) and/or client-machines accessing specific content being delivered
17 by their server(s) in a network environment. Such client machines may be in the form of
18 desktop or laptop personal computers, pocket computers, personal digital assistants, cellular
19 telephones, cellular pagers, or any other electronic devices capable of communicating data over
20 a network. The present invention is ideally suited for protecting subscription-type Web sites
21 (such as adult content sites), on-line banking sites, and corporate intranets that allow remote
22 access by tele-commuters and travelers. Fig. 1 broadly illustrates such a system wherein three
23 different client machines 10, 12, and 14 are interconnected with a central server computer 16 by
24 network 18. Network 18 can be a local area network; alternatively, it might represent a dial-up
25 connection and/or a global computer network (wide area network) such as the Internet or a
26 network similar thereto; network 18 could be a wired network or a wireless network.

27 In order to make use of the present invention, the server administrator installs a special
28 server application on the server computer. This server application is set to automatically launch

1 when the administrator's server machine is restarted. The server administrator also installs a
2 programming code library (such as a dynamic link library, or DLL, in the Windows®
3 environment) that is used to query the temporary storage table containing session identifiers
4 (and other information) for authorized client machines. This code can be embedded within a
5 protected Web page. A related grouping, or "content cluster", of such Web pages share a unique
6 DLL defining such "cluster". This DLL thereby serves as a "key" specific to each such cluster.
7 Two or more such DLLs can be used to distinguish between two or more clusters (e.g., two or
8 more databases), offered on the same Web site. Similarly, two or more such DLLs can be used
9 to distinguish between two sets of subscribers having the same access level but different
10 subscription expiration dates. For instance, the programming code library embedded into each
11 page of a protected cluster could be an ActiveX Dynamic Link Library (DLL) Object designed
12 for an Active Server Page (ASP) page being served by a Microsoft Internet Information Server
13 (IIS®). While the present invention can be used successfully with a host of operating systems,
14 servers, and network environments, the preferred embodiment makes use of a Microsoft®
15 Internet Information Server (IIS) Web server running Microsoft® Windows® being accessed by a
16 user with a computer (client machine) running Microsoft® Windows® and a Java®-enabled Web
17 browser.

18 A small "client-side" software application, or client tray application, must be installed
19 and run on each client machine that is to have access to protected Web pages maintained by
20 server 16 before such client machine can access protected Web content. Referring to the
21 flowchart of Figs. 2A, 2B, and 2C (hereinafter, "the Fig. 2 flowchart"), control starts at block 20
22 and passes to a decision diamond 22 to determine whether the aforementioned client tray
23 application has already been installed on the user's client machine. If not, then a request is
24 preferably communicated over the network by the client machine to the server for a download of
25 the client tray application software, represented by box 24 in Fig. 2A. Incidentally, within the
26 Fig. 2 flowchart, vertical dashed line 19 symbolically represents the division between a client
27 machine (the process steps for which are shown on the left side of such dashed line) and the
28 server computer (the process steps for which are shown on the rights side of such dashed line).

1 It is not necessary that the client tray application be downloaded from the same server computer
2 that maintains the protected Web pages; indeed, the client tray application could be obtained by
3 a user from an entirely different computer by downloading or by email attachment, or via a
4 CDROM, DVD, floppy disk, etc., mailed out by the operator of the server computer. Once the
5 client tray application is downloaded or otherwise acquired by the user, the client tray
6 application is installed on the user's client machine, as indicated by block 26 in Fig. 2A.
7 Preferably, the client tray application is installed in a manner by which it is automatically
8 launched in the background and remains resident within the memory of the client machine each
9 time the client machine is "booted up". The user need only install the client tray application if
10 it has not been previously installed. The same client tray application can be used to access
11 content on any server protected by the method of the present invention. Thus, a given client
12 machine, authorized to access two or more different subscription services, could make use of the
13 same client-side software on the client machine to control access to each of such services.
14 However, for each different service, the user would be issued a separate password, as described
15 in greater detail below.

16 As mentioned above, an initial request for access to any IIS[®]-ASP-enabled Web page
17 causes a unique session identifier to be created and returned to the user's client machine as a
18 "cookie". An "entry Web page" on the server containing protected content includes software
19 code, such as a Java[®] Applet, that is automatically downloaded to a client user's machine when
20 access to such entry Web page is requested for the first time during the user's current session.
21 This Java[®] applet is executed by the Web browser of the client user's machine with or without
22 user intervention, and is either: 1) "digitally signed", or 2) set as originating from a "trusted"
23 source upon the first visit made by any browser client application to each protected
24 server/domain, as is the usual process for any Java[®] applet performing advanced client-machine
25 processes. The Java[®] applet couples the client tray application (already running in the
26 background on the user's client machine) with the aforementioned server application. The
27 Java[®] applet communicates with the client-side tray application (which, in turn, communicates
28 with the client machine key DLL described below) in order to determine whether this client

1 machine is authorized to access any of the protected clusters maintained on this Web site. For
2 example, the tray application might return an integer ranging from 0 through 10 (the "Integer
3 Value"), wherein the value of the integer indicates what level of access is to be provided,
4 whether the user's subscription has expired ("client expired"), etc. The Java® applet then
5 updates the temporary storage table that contains valid session identifiers, assuming that access
6 is to be granted. In other words, the temporary storage table stores information only for
7 validated client machines. The temporary storage table preferably (at minimum) stores the
8 Cluster Number, Session I.D. (Session identifier), Integer Value, date-time stamp (to facilitate
9 purging of older/expired temporary records from the temporary storage table), and "anti-
10 spoofing" checksum value(s) (including but not limited to client-machine calculated checksums
11 calculated on 1) a date-time specific password string (to be verified by comparing the client-
12 machine calculated checksum with a server calculated checksum calculated on the expected
13 date-time specific password for the date-time that the Java Applet was served to the client), 2)
14 the tray application executable file on the client-machine). Since all further access requests for
15 Web pages hosted by this server during a particular Web browsing session automatically include
16 the initial session identifier, the server can check the temporary storage table without requiring
17 the Java® applet to execute, and without any further communication between the Java® applet
18 and the tray application. This specific Java®-based method minimizes firewall and proxy server
19 problems that an end-user might otherwise experience if the tray application was designed to
20 communicate directly with a remote server.

21
22
23 Each protected Web site (or each protected "content cluster" of each Web site) has a
24 unique dynamic link library (DLL) associated therewith. This DLL is sometimes referred to
25 herein as a "client machine key", and it is this DLL that generates the machine-specific identifier
26 for the user's client machine. This client machine key DLL provides a virtually unique client
27
28

1 machine identifier, and includes an algorithm for ensuring that the user enters a password
2 uniquely corresponding to the unique client machine identifier before access to protected data is
3 granted. If a user desires access to such protected Web site, or to the protected “content
4 cluster”, then the specific Web site DLL must exist on the user’s client machine. It should be
5 noted that the client tray application serves to couple the Java® applet and the client machine key
6 DLL. The client tray application detects whether the Web site DLL exists on the client machine,
7 as indicated by decision diamond 30. If not, then a request is made to the Web site server, as
8 represented by block 32, that the Web site DLL be downloaded to the client machine. The Web
9 site server thereafter downloads such DLL to the client machine, and such DLL is installed by
10 the client tray application, as indicated by block 34. The server administrator can group
11 protected content into different “clusters” and use a different DLL, or client machine key, for
12 each protected content cluster, if desired, enabling the Web administrator to use the present
13 invention to perform multi-level access control. However, as described above, multi-level
14 access control can also be provided by using the integer value returned by the client machine
15 key DLL via the client tray application.

16
17
18
19 The client tray application, together with the client machine key DLL, determines the
20 identity of the client-machine and other unique information including the “authorization
21 rights/status of the client-machine” relative to the current server or content cluster being
22 accessed. The step of generating machine-specific identification is represented in Fig. 2A by
23 block 38. In the preferred embodiment, this step is achieved by using a software application
24 package known as SoftSentry software, or a similar application, in conjunction with the client
25 tray application. The client machine key DLL (typically generated by the Web site administrator
26 using SoftSentry software) analyzes hardware characteristics of a particular local computer, or
27
28

1 client machine (including hard drive characteristics, RAM characteristics, input/output device
2 parameters and other hardware specific details), and thereby generates an integer value, or
3 machine-specific identifier that is virtually unique for a given client machine. The client
4 machine key DLL then performs an algorithm to calculate a password, based upon the machine-
5 specific identifier, valid only for a client machine having the same virtually unique system
6 identifier. As a result, even if the client tray application (and/or client machine key DLL) is
7 copied by a user to a second client machine, the client machine key DLL on the second client
8 machine will report a new machine-specific identifier that is almost certainly different from the
9 machine-specific identifier that was generated by the user's first client machine.
10

11
12 The first time that the client machine key DLL for the Web site currently being accessed
13 is queried by the system tray application, the client tray application needs to transmit the
14 machine-specific identifier (computed by the client machine key DLL) to the server as indicated
15 by arrow 42 in Fig. 2A, in order to request a password. The request for a password can be done
16 manually by the user by telephone, telefax, email etc. Alternatively, the request for a password
17 can be transmitted directly over the network. Likewise, the assignment of such password can be
18 handled automatically by the server side application software, without human intervention, and
19 transmitted over the network directly to the client machine; alternatively, the password can be
20 provided by a human system administrator to the user via telephone, email, telefax, by mail, or
21 by similar means. In either case, the server side application software and/or human
22 administrator of the server system applies an algorithm to the machine-specific identifier
23 received from the client machine and/or client machine user to compute a password for such
24 client machine, as illustrated by block 46 in Fig. 2B.
25
26

27 In one preferred embodiment of the present invention, the server application software
28

1 issues passwords to client machines in the following manner: A user who wishes access to
2 protected data will visit a Web page (i.e., a Web page maintained by a server running the server
3 side application software), as by entering a URL directly into the user's Web browser, via
4 clicking on a Web link, via clicking on a "subscribe now" button on the client-machine key DLL
5 user interface, etc., for allowing a user to place an online "order" for access to the protected data
6 maintained by such server. The user manually enters the user's credit card information and
7 email address for transmission to the server side application software. In addition, the user
8 transmits to the server side application software the user's client-machine system identifier
9 generated in block 38 of Fig. 2A, as by clicking on a "button" displayed by the client-machine
10 key DLL user interface, for instance; of course, this step is not necessary if the user's client-
11 machine system identifier was previously passed to the server side application software during
12 the current browsing session (manually or automatically). Upon credit-card processing
13 approval, a client-machine key DLL password is transmitted by the server side application
14 software, via email, to the user. In some cases, the ordering process may allow the user to select
15 from a variety of subscription lengths and/or access levels (standard, advanced, etc.),
16 presumably with different prices associated with each. The particular options selected by the
17 user then dictate the specific client-machine key DLL password that is issued to such user.

18
19
20
21 A user is then prompted by the client machine key DLL to enter a password into the
22 client machine, which, if valid, is saved for future reference by the aforementioned site/cluster-
23 specific, client machine key DLL, as indicated by block 50 in Fig. 2B. If a valid password
24 (discussed below) is not entered, then future attempts to access such Web site are denied
25 pending entry of a valid password.
26

27 At this point, it might be noted that the client machine key DLL could be configured to
28

1 accept two or more alternate passwords for a given protected site, Web page, or cluster.

2 Different passwords could be used to convey different subscription terms (one year, two years,
3 etc.) or different access levels. The server administrator, or the automated payment/password
4 issuing system (described above) selects the appropriate password to be conveyed to the user.
5

6 The next step in the validation process is for the client machine key DLL to compare the
7 password computed by the client machine key DLL in step 38 with the password information
8 entered by the user in step 50, as indicated by decision diamond 52. If the validation fails, then
9 access to the protected Web page is denied, as indicated by termination step 54. However, if the
10 validation is successful, then the aforementioned Java[®] applet transmits the aforementioned
11 session i.d. and authorized access level to the temporary storage table, located on a remote
12 server, over the network, as indicated by block 58 in Fig. 2B. The server on which the
13 temporary storage table is maintained (this need not be the same server as the server which
14 serves the protected Web pages) has an application program (referred to herein as the
15 “temporary storage table application”) installed and running thereon which controls reading
16 from, and writing to, such temporary storage table.
17
18

19 As mentioned above, the temporary storage table application maintains a temporary data
20 table of authorized, logged-on users, including, among other things, any related access level
21 information for such users. When a validated client machine transmits its session identifier and
22 access level to the temporary storage table application (see step 58), this data is temporarily
23 saved in the temporary storage table, as indicated by block 60. The temporary storage table
24 application performs local checks of the data received from the client tray application to
25 determine that it is originating from the authorized client-machine, preventing the insertion of
26 “spoof” client-machine authorizations into the server application’s temporary database.
27
28

1 Following storage of the client-machine authentication data in the temporary storage table,
2 future requests by the client for file/data access can easily be processed because each such
3 file/data request incorporates this session identifier information.

4
5 When a user requests remote files via a Web browser, the user's browser on the client
6 machine sends HTTP header data taken from a browser "cookie" that contains the session i.d.
7 information to the Web server as a means of identifying the client machine, as indicated by
8 block 64 and arrow 66. The user's browser also specifies the address/file name of the secured
9 page that the user is attempting to access. The secured page's code now interacts with the
10 ActiveX Dynamic Link Library (ActiveX DLL) Object embedded in the secured Web page on
11 the server. The code retrieves the session identifier information from the header included in the
12 browser's "GET" request and then passes this session identifier information to the ActiveX
13 DLL. The ActiveX DLL then prompts the temporary storage table application to search for
14 matching session i.d. information, as designated by decision diamond 68. The temporary
15 storage table application searches the temporary storage table for temporarily stored
16 authentication data matching the passed session identifier information, and then responds with
17 an "authorize client message," "deny client" message, "expired client" message and/or access-
18 level data, and the ActiveX DLL then passes such response(s) to the secured page's code. If the
19 secured page's code receives an indication that a valid/authorized session identifier was found
20 (and that appropriate access level rights are indicated, if applicable), as represented by arrow 72,
21 then the server login/connection, or content request, is granted, and the requested content is
22 served to the user, as indicated by blocks 74 and 76. On the other hand, if the secured page's
23 code receives an indication that: 1) the session identifier was not found in the temporary storage
24 table; 2) the user is attempting to access a page beyond the scope of the user's access level
25
26
27
28

1 rights; or 3) the user's subscription has expired, then the client-machine, and therefore the user,
2 are denied access to the server or secured content, as represented by termination block 70. In
3 this case, the client machine's browser is redirected to a notification page, where the user
4 receives information about why access was denied.
5

6 If desired, custom code can be incorporated into the server application software to query
7 a user database or other objects. This custom code can be used to control such things as
8 subscription expirations, user authentication passwords, and the like by nesting "if...then"
9 statements outside of the secured page's code. In other words, the present invention can
10 optionally be coupled with user authentication/security technologies to provide additional levels
11 of security. If an administrator couples user authentication with the present invention, then
12 stolen authorized client-machines and unauthorized users with access to authorized client-
13 machines do not present a security threat.
14

15 Moreover, it will be noted that the temporary storage table application software is multi-
16 thread capable, and maintains a list of temporarily authorized clients. Multiple client-
17 side/server-side combinations may be running simultaneously on any user/server machine. In
18 other words, a user can have multiple simultaneous "client machine keys" installed on his
19 computer, allowing access to one or more protected Web sites and/or Web pages ("clusters") at
20 any given time. In the same spirit, a Web server can host multiple unique server data "locks,"
21 acting as a sentry to multiple unique Web sites/Web pages ("clusters") simultaneously. As
22 mentioned above, the same client-side tray application software can be used to control access by
23 a particular client machine to one or more protected sites and/or services. The client machine
24 simply has one or more client machine key DLLs, and the client machine is issued a unique
25 password for each such client machine key DLL, corresponding to each such protected site
26
27
28

1 and/or service ("cluster").

2 It will also be noted that the present invention does not require a pre-populated database
3 of any kind on any server. In contrast, most user authentication technologies require constant
4 administration of a user database to control access to a server and/or server resources. Instead,
5 the present invention uses dynamic information from the client-machine to automatically insert
6 "temporary records" into an administration-free database table on a server storing a list of
7 temporarily authorized client-machines.
8

9 The present invention can be used with any type of third-party server (WWW, FTP,
10 mail, database, telnet, gopher, etc.) capable of tracking session identifier information and
11 processing custom code. Because the present invention works as an add-on/plug-in to third-
12 party servers, server administrators and users do not lose any of the functionality or features of
13 the server being used to serve the data/content to the user. This is in contrast to many security
14 solutions that employ a proprietary server to serve secured content to the end-user. Web page
15 data protected by the present invention can be accessed by any authorized client-machine using
16 browser software capable of communicating with the aforementioned client tray application, all
17 without any need for additional coupling tools. For example, the aforementioned Java[®] applet
18 facilitates any Java[®] 1.1 capable Web browser's communication with the client tray application.
19 Some types of network communication client applications (including FTP, mail, database,
20 telnet, gopher, etc. but not necessarily including Java 1.1 capable web browsers) may require
21 additional coupling tools (plug-ins or patches for example) to leverage the security of the
22 present invention. Custom network communication client applications (including WWW, FTP,
23 mail, database, telnet, gopher, etc.) could also be developed specifically designed to leverage the
24 security of the present invention all without any need for additional coupling tools.
25
26
27
28

1 The present invention is ideally suited to be used by Web server administrators having
2 Internet or Intranet Web pages/sites who require total control over access to a company's Web
3 servers and/or Web services. It provides greater security than known hardware-lock (or dongle)
4 type systems, since it limits access to specific client machines. Accordingly, it is ideally suited
5 for use with subscription-type Web sites, on-line banking sites, and for remote logins by
6 travelers and telecommuters to corporate intranets. Under conditions of solid network
7 connectivity, the entire authentication process practiced by the present invention executes very
8 quickly.
9

10
11 Those skilled in the art will note that there are differences between Web page access and
12 FTP file transfer service. Web pages make use of a "stateless" protocol wherein the Web
13 browser on the client machine is not connected continuously to the server for the entire duration
14 of a browsing session. For example, when a remote user first connects to a Web page served by
15 a Microsoft Internet Information Server (IIS®) software, the remote user's browser connects
16 with the home page of the addressed site, gets the page, then disconnects from such site. When
17 the remote users click on a link on such home page, the remote user's browser connects with the
18 linked Web page, gets the page, then disconnects again. The session identifier and temporary
19 storage table are used to track and authenticate each of those connections. In contrast, an FTP
20 Client is connected to the FTP Server during the entire session. Authentication need only be
21 done at the beginning of the session, and hence, there is no need for a temporary storage table to
22 temporarily save authenticated browsing session identifiers. In the FTP protocol environment,
23 all that is needed is an FTP client and an FTP server; the FTP client needs to know how to
24 communicate directly with the client machine tray application. It will be recalled from above
25 that Java® Applets can be used for such purpose in a Web browser environment, but Java®
26
27
28

1 Applets can only be run in a web browser, and not in an FTP client. Likewise, the FTP server
2 needs to know how to communicate with the FTP client to authenticate the client machine.
3 However, by making use of the present invention in the FTP environment, the user can avoid the
4 need to use a username and password to log into the FTP server, as they typically do now.
5

6 Those skilled in the art will now appreciate that a novel and improved client/server
7 authentication system has been described which provides an increased level of security, without
8 requiring a user to possess any special hardware components, or “dongles”, in order to gain
9 access to the server. Moreover, the present invention serves to restrict a client to a particular
10 machine in order to gain access to a Web site and/or Web pages on an Internet and/or intranet
11 site. In addition, the authentication system of the present invention easily allows for multiple
12 levels of access wherein some authorized users have greater access to Web pages than other
13 authorized users. All of these advantages are provided without the disadvantage of burdening
14 the server computer with the need to maintain a pre-populated database of authorized users.
15

16 While the present invention has been described with respect to preferred embodiments thereof,
17 such description is for illustrative purposes only, and is not to be construed as limiting the scope
18 of the invention. Various modifications and changes may be made to the described
19 embodiments by those skilled in the art without departing from the true spirit and scope of the
20 invention as defined by the appended claims.
21
22
23
24
25
26
27
28

1 We claim:

2 1. A method of restricting access to data maintained on a server computer by an
3 authorized client machine, said method comprising the steps of:

4 a. installing a client-side software program on the client machine for generating a client
5 machine-specific identifier, the client machine-specific identifier being unique to the particular
6 machine upon which such client-side software program is initially installed;

7 b. operating the client-side software program on the client machine to generate the
8 client machine-specific identifier;

9 c. generating a unique password remote from the client machine, and providing the
10 unique password to a user of the client machine, the unique password being derived from the
11 client machine-specific identifier generated in step b., and uniquely corresponding thereto;

12 d. issuing a request by the client machine to the server computer for access to data
13 maintained on the server computer;

14 e. responding to the request for access of step d. by having the client machine re-
15 generate its machine-specific identifier;

16 f. verifying whether the client machine-specific identifier re-generated in step e.
17 uniquely corresponds with the unique password generated in step c.; and

18 g. recognizing the client machine as being authorized to access data maintained on the
19 server computer if the verification performed by step f. is true, and refusing to recognize the
20 client machine as being authorized to access data maintained on the server computer if the
21 verification performed by step f. is false.
22
23
24
25
26
27
28

1 2. The method recited by claim 1 wherein the verification step f. of claim 1 is
2 performed by the client machine.

3
4
5 3. The method recited by claim 1 wherein the verification step f. of claim 1 is
6 performed by a server computer remote from the client machine.

7
8 4. The method recited by claim 3 wherein the verification step f. of claim 1 is
9 performed by the server computer that maintains the data which the client machine is trying to
10 access.
11

12
13 5. The method recited by claim 1 including the step of installing a server-side software
14 program on a server computer, remote from the client machine, for generating the unique
15 password generated in step c. of claim 1.
16

17
18 6. The method recited by claim 5 wherein the server-side software program is installed
19 on the server computer that maintains the data which the client machine is trying to access.
20

21 7. The method recited by claim 1 wherein the step of generating a unique password
22 remote from the client machine is performed by a computer other than the client machine.
23

24
25 8. The method recited by claim 1 wherein the step of installing the client-side software
26 program on the client machine includes the step of downloading the client-side software
27 program from the server computer.
28

1 9. The method recited by claim 1 wherein the step of generating a unique password
2 remote from the client machine includes the step of transmitting the client machine-specific
3 identifier generated in step b. to a password-generating computer other than the client machine,
4 and generating the unique password at the password-generating computer.
5

6
7 10. The method recited by claim 9 wherein the password-generating computer is the
8 server computer that maintains the data which the client machine is trying to access.
9

10 11. The method recited by claim 9 wherein the password-generating computer operates a
11 software program to perform an algorithm for generating the unique password based upon the
12 client machine-specific identifier generated in step a.
13

14
15 12. The method recited by claim 1 further including the steps of:
16

17 h. obtaining a session identifier and a site identifier from the client machine each time
18 the client machine requests access to restricted data area of the server computer, the session
19 identifier indicating a particular working session by the client machine, and the site identifier
20 indicating a particular data area of the server computer which the client machine is authorized to
21 access;
22

23 i. storing the session identifier and site identifier in a temporary storage table remote
24 from the client machine upon a first request by the client machine for access to a restricted data
25 area of the server computer;

26 j. comparing the session identifier obtained from the client machine upon subsequent
27 requests for access to the restricted data area of the server following the first request; and
28

1 k. allowing the client machine continued access to the restricted data area of the server
2 computer if the session identifier obtained from the client machine upon a subsequent request
3 corresponds to a session identifier already stored in the temporary storage table.
4

5
6 13. The method recited by claim 1 wherein each client machine is only authorized to
7 access particular data areas of the server computer, and wherein the client-side software
8 program installed on the client machine determines which areas of the server computer can be
9 accessed by a particular client machine.
10

11
12 14. A method of restricting access to data maintained on a server computer by an
13 authorized client machine, said method comprising the steps of:

14 a. creating a session identifier in a computer remote from the client machine for a
15 current browsing session of the client machine;

16 b. transmitting to the client machine the session identifier created in step a);

17 c. storing the session identifier transmitted in step b) within the client machine;

18 d. verifying that the client machine is authorized to access data maintained on the
19 server computer;
20

21 e. obtaining the session identifier stored in step c), and storing such session identifier
22 within a storage table remote from the client machine if such client machine was verified in step
23 d);
24

25 f. transmitting a request by the client machine for access to data maintained on the
26 server computer, such request including the session identifier stored in step c);

27 g. comparing the session identifier transmitted in step f) with the session identifier
28

1 stored in the storage table during step e) to determine whether the request for access transmitted
2 in step f) is authorized; and
3 h. permitting access by the client machine to the requested data maintained on the
4 server computer if the comparison made in step g) shows that the request for access is
5 authorized, and denying access by the client machine to the requested data maintained on the
6 server computer if the comparison made in step g) shows that the request for access is not
7 authorized.
8

9
10 15. The method recited by claim 14 wherein the session identifier stored on the client
11 machine in step c) is stored as a temporary file on the client machine.
12

13
14 16. The method recited by claim 15 wherein the temporary file which stores the session
15 identifier on the client machine is a "cookie".
16
17
18
19
20
21
22
23
24
25
26
27
28

Abstract of the Disclosure

A method of restricting access to data maintained on a server computer by one or more authorized, networked client machines includes the step of installing a client-side software program on the client machine for generating a client machine-specific identifier determined by particular characteristics of the client machine. The machine-specific identifier is used by the server administrator to generate a unique password, which the user enters into the client-side software program. Server-side software is embedded on protected Web pages of the server computer that hosts the protected Web site. When a user desires access to protected content, the client-side software is prompted to re-generate its machine-specific identifier and valid client password list for comparison with the password previously entered by the user. Access is granted if they correspond, and denied if they do not. If the client machine is recognized as being authorized to access data on the protected Web site during a first access request, then the current session identifier is saved in a temporary storage table remote from the client machine for indicating current working sessions of authorized client machines. The client machine returns such session identifier with each additional request for access, and the temporary storage table is consulted to search for such session identifier before granting access.

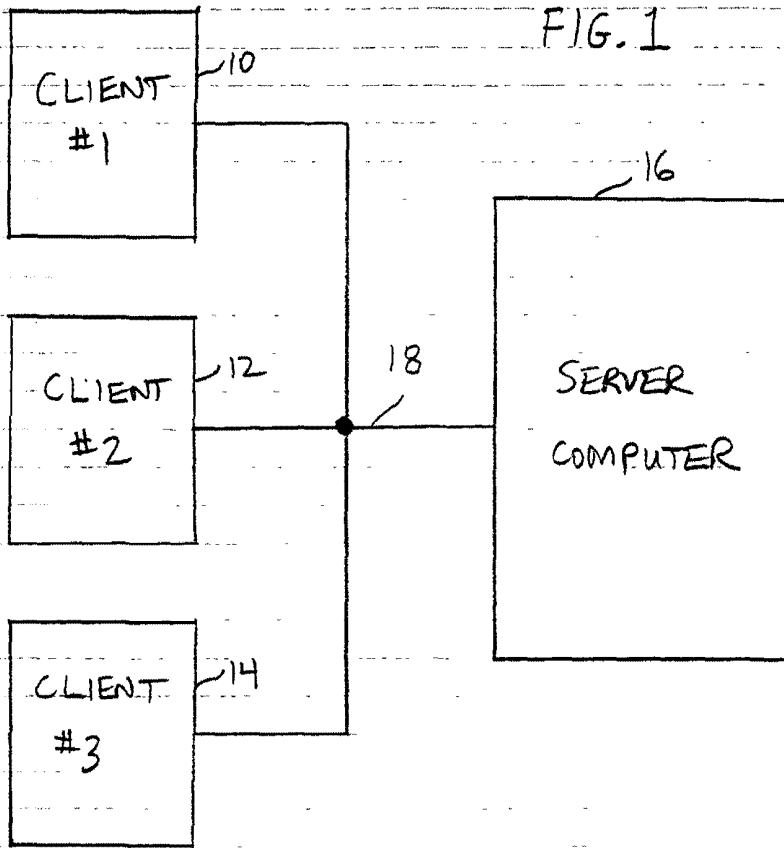


FIG. 2A

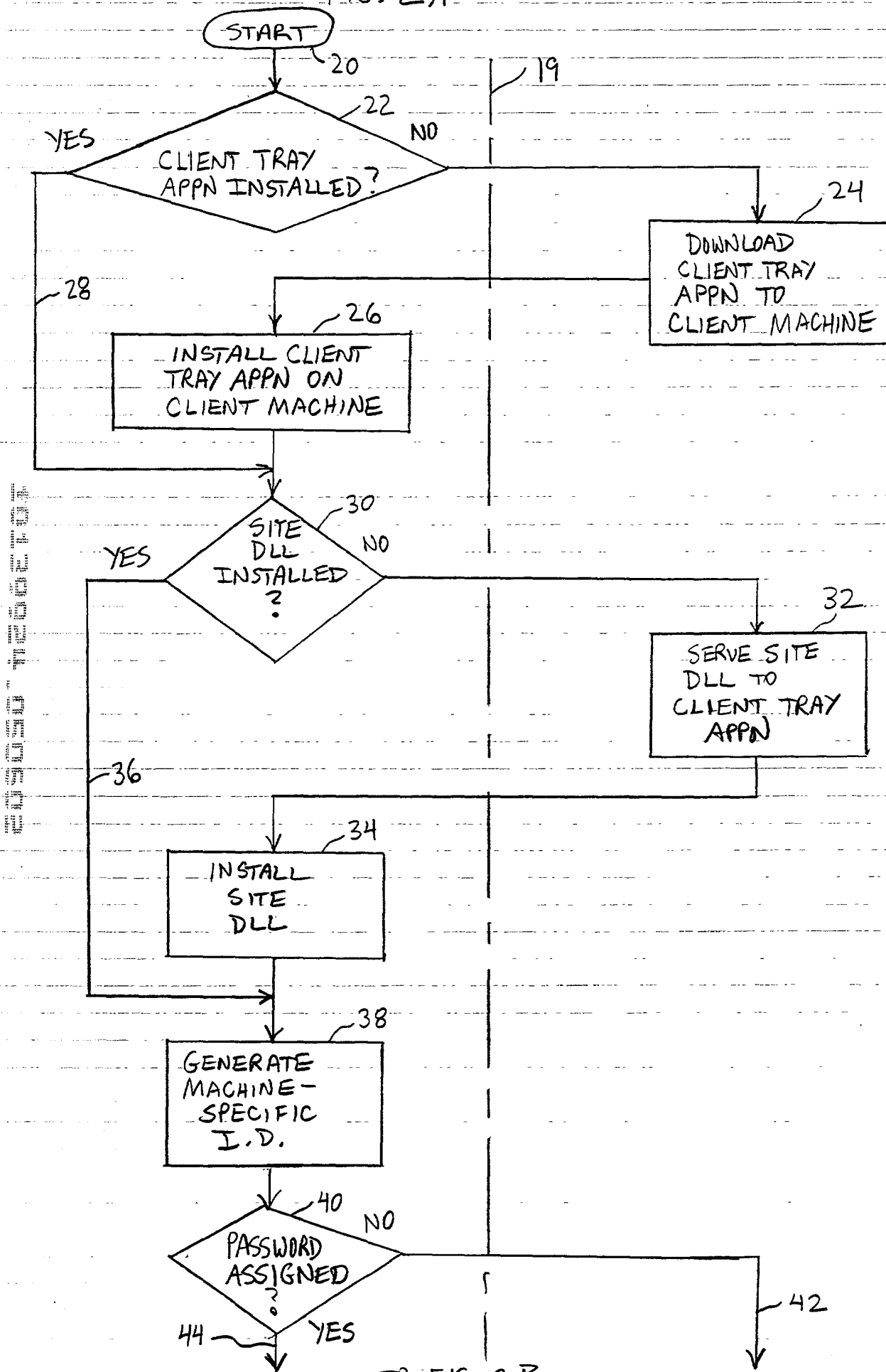
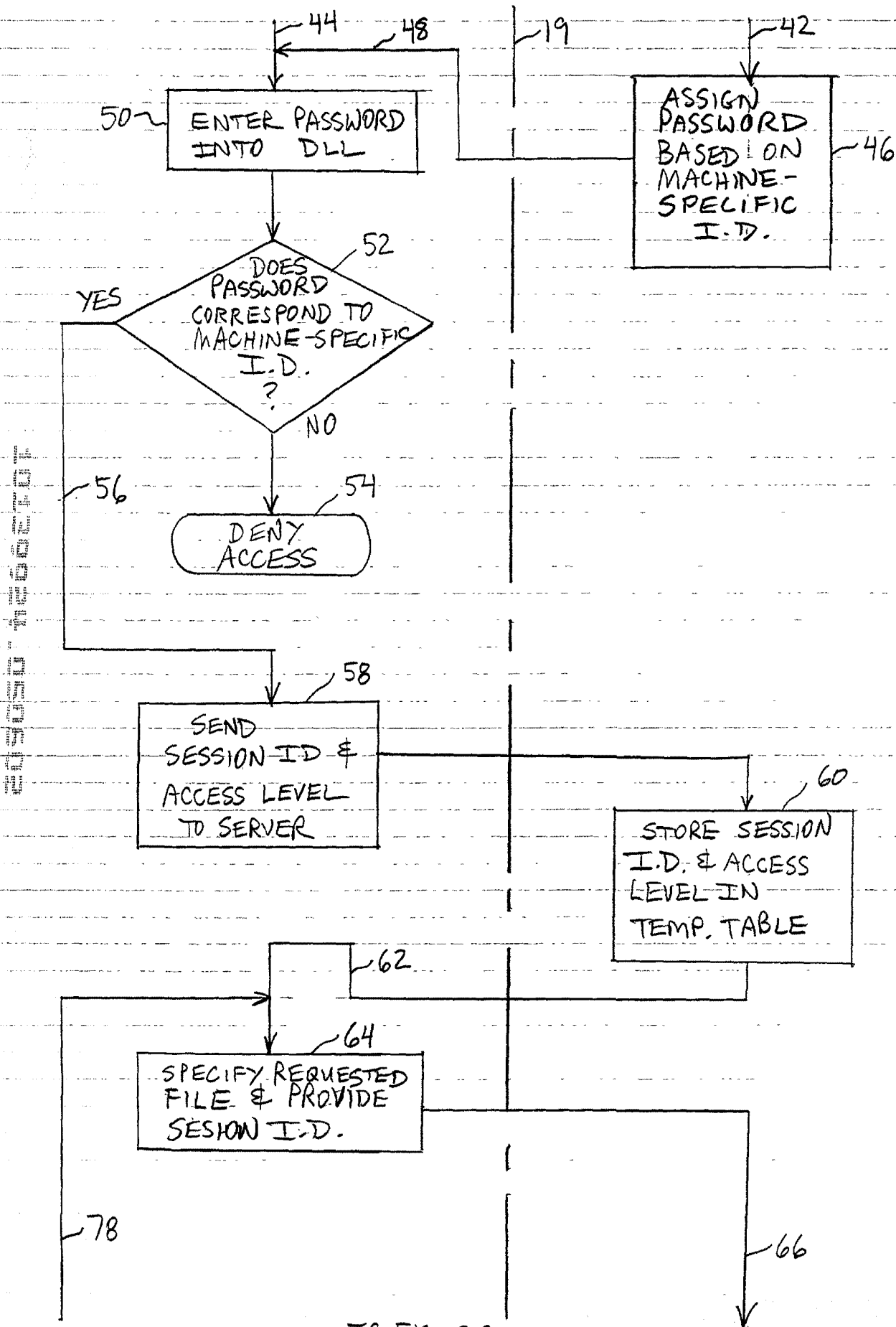
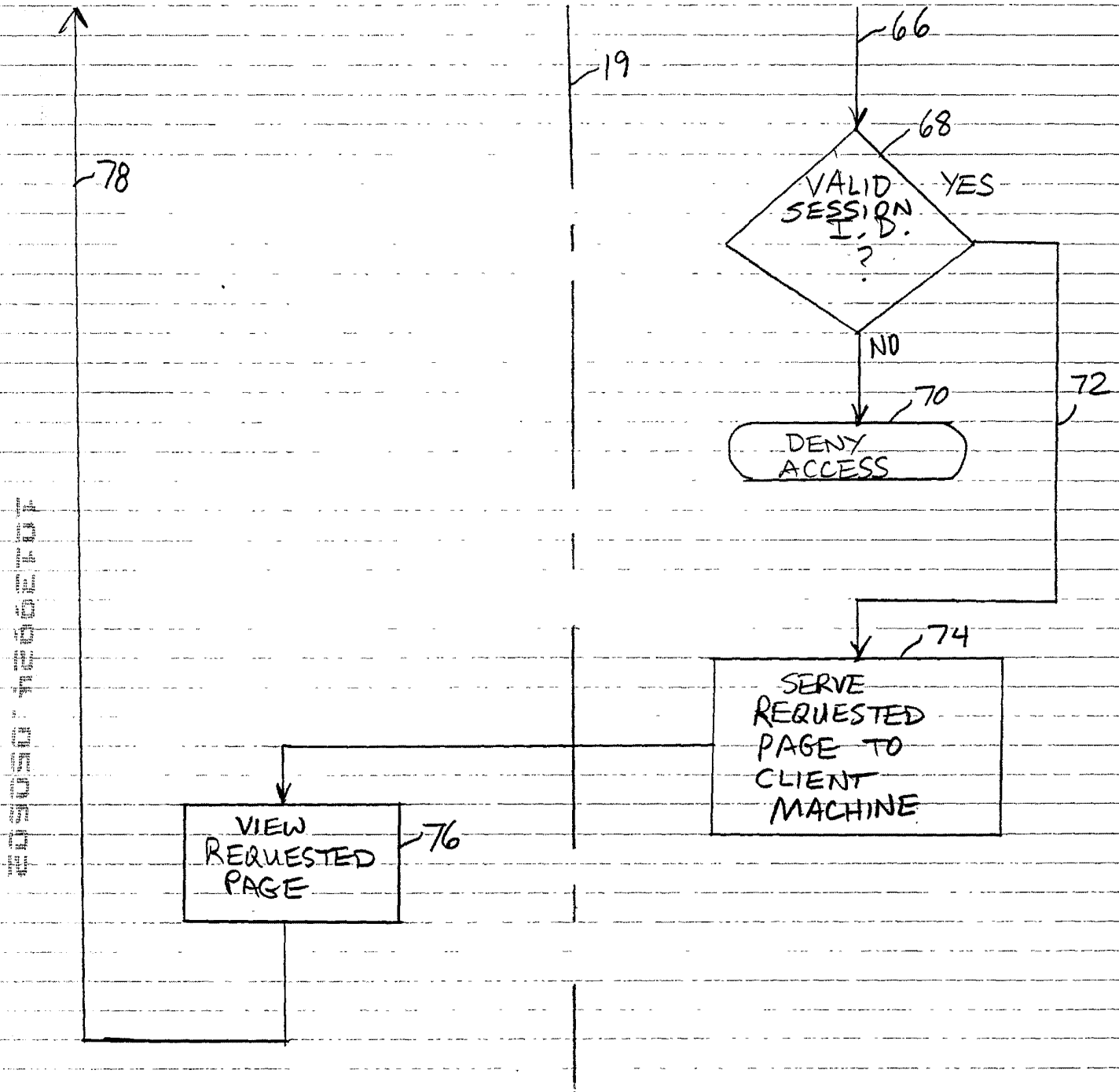


FIG. 2B



TO FIG. 2C

FIG. 2C



DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am an original, first and joint inventor of the subject matter which is claimed in an application entitled "**METHOD FOR RESTRICTING ACCESS TO A WEB SITE BY REMOTE USERS**"

the specification of which X is attached hereto. ____ was filed on _____ and assigned Application Serial No. _____ and was amended on _____.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the Patent Office all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)

<u>Not applicable</u>			<u>Priority claimed</u>	
<u>(Number)</u>	<u>(Country)</u>	<u>(Date filed)</u>	<u>Yes</u>	<u>No</u>

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

And I hereby appoint:

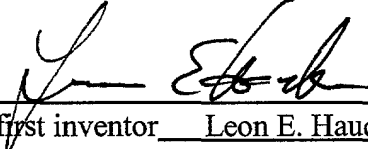
William C. Cahill, Reg. No. 19,742; C. Robert von Hellens, Reg. No. 25,714;
Charles R. Hoffman, Reg. No. 26,556; Marvin A. Glazer, Reg. No. 28,801; and
Thomas G. Watkins III, Reg. No. 27,964;


my attorneys with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith.

Address all telephone calls to: Marvin A. Glazer, (602) 956-7000

Address all correspondence to: CAHILL, SUTTON & THOMAS P.L.C.
Attn: Marvin A. Glazer
155 Park One
2141 East Highland Avenue
Phoenix, Arizona 85016

Wherefore I request that Letters Patent be granted to me for the invention or discovery described and claimed in the above-identified patent application.

Inventor's signature  Date 5/6/02
Full name of sole or first inventor Leon E. Hauck
Residence 4613 W. Butler Drive, Glendale, AZ 85302
Post Office Address 4613 W. Butler Drive, Glendale, AZ 85302
Citizenship U.S.

Inventor's signature  Date 5/6/02
Full name of joint inventor Brent J. Burval
Residence 16614 S. 21st Street, Phoenix, AZ 85048
Post Office Address 16614 S. 21st Street, Phoenix, AZ 85048
Citizenship U.S.

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

05/09/2002 SDIRETA1 00000027 10139924

01 FC:201

370.00 OP

PATENT APPLICATION FEE DETERMINATION RECORD

Effective October 1, 2001

Application or Docket Number

10139924

CLAIMS AS FILED - PART I

(Column 1)

(Column 2)

TOTAL CLAIMS	16	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	16 minus 20 = *	—
INDEPENDENT CLAIMS	2 minus 3 = *	—
MULTIPLE DEPENDENT CLAIM PRESENT <input type="checkbox"/>		

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

(Column 1)

(Column 2)

(Column 3)

AMENDMENT A		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	*	Minus	**	=
	Independent	*	Minus	***	=
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM <input type="checkbox"/>				

SMALL ENTITY TYPE ☐

OR

OTHER THAN SMALL ENTITY

RATE	FEE		RATE	FEE
BASIC FEE	370.00	OR	BASIC FEE	740.00
X\$ 9=		OR	X\$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL	370	OR	TOTAL	

SMALL ENTITY

OR

OTHER THAN SMALL ENTITY

RATE	ADDI-TIONAL FEE		RATE	ADDI-TIONAL FEE
X\$ 9=		OR	X\$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL		OR	TOTAL	
ADDIT. FEE		OR	ADDIT. FEE	

(Column 1)

(Column 2)

(Column 3)

AMENDMENT B		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	*	Minus	**	=
	Independent	*	Minus	***	=
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM <input type="checkbox"/>				

RATE	ADDI-TIONAL FEE		RATE	ADDI-TIONAL FEE
X\$ 9=		OR	X\$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL		OR	TOTAL	
ADDIT. FEE		OR	ADDIT. FEE	

(Column 1)

(Column 2)

(Column 3)

AMENDMENT C		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	*	Minus	**	=
	Independent	*	Minus	***	=
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM <input type="checkbox"/>				

RATE	ADDI-TIONAL FEE		RATE	ADDI-TIONAL FEE
X\$ 9=		OR	X\$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL		OR	TOTAL	
ADDIT. FEE		OR	ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

CLAIMS ONLY							SERIAL NO.	FILING DATE
							APPLICANT(S)	
CLAIMS								
	AS FILED		AFTER 1st AMENDMENT		AFTER 2nd AMENDMENT			
	IND.	DEP.	IND.	DEP.	IND.	DEP.		
1							51	
2		/					52	
3		/					53	
4		/					54	
5		/					55	
6		/					56	
7		/					57	
8		/					58	
9		/					59	
10		/					60	
11		/					61	
12		/					62	
13		/					63	
14	/						64	
15		/					65	
16		/					66	
17							67	
18							68	
19							69	
20							70	
21							71	
22							72	
23							73	
24							74	
25							75	
26							76	
27							77	
28							78	
29							79	
30							80	
31							81	
32							82	
33							83	
34							84	
35							85	
36							86	
37							87	
38							88	
39							89	
40							90	
41							91	
42							92	
43							93	
44							94	
45							95	
46							96	
47							97	
48							98	
49							99	
50							100	
TOTAL IND.	2	↓		↓		↓	TOTAL IND.	↓
TOTAL DEP.	14	←		←		←	TOTAL DEP.	←
TOTAL CLAIMS	16						TOTAL CLAIMS	

* MAY BE USED FOR ADDITIONAL CLAIMS OR ADMENDMENTS

FORM PTO-2022 (1-98)

U.S. DEPARTMENT OF COMMERCE
Patent and Trademark Office